STABLE USER-SPECIFIC HAPTIC RENDERING OF THE VIRTUAL WALL

R. Brent Gillespie

Department of Mechanical Engineering
Northwestern University
Evanston, Illinois 60208, USA
b-gillespie@nwu.edu

Mark R. Cutkosky

Department of Mechanical Engineering
Stanford University
Stanford, California 94305, USA
cutkosky@cdr.stanford.edu

ABSTRACT

Efficient control algorithms are developed to implement stiff virtual walls without chatter. An analysis of the complete coupled system comprising controller, interface device, and user's finger underlies the design of a wall algorithm, thus each virtual wall is tailored to a specific user impedance. The finger is modeled as a static second order impedance with justification drawn from empirical studies of limb dynamics available in the literature and from observations of the disparity in time scales between contact instability and volitional control. Compensation is incorporated for the destabilizing effects of the zero order hold using either model based prediction or design in the digital domain. The destabilizing effects of asynchronous wall on/off switching times and sampling times are tracked by a special watchdog while deadbeat control is used to periodically eliminate these effects. Extensions are discussed, including on-the-fly system identification of the user impedance.

INTRODUCTION

Despite its apparent simplicity and its status as a fundamental building block of most virtual objects, the virtual wall often evades perceptually convincing renderings. When touching a wall, especially one which is meant to be stiff, meddlesome oscillatory motion of the manipulandum (often called contact instability or chatter) tends to arise. Such behavior immediately obliterates any sense of immersion which the user may have been enjoying: this non-passive behavior is not typical of real world experience. These sustained oscillations can be attributed to the introduction of mechanical energy into the coupled dynamical system (simulated wall, interface device, and human limb) through "energy leaks". This paper addresses certain energy leaks which arise through the coupling of a wall which is simulated in discrete time to a physical device and human limb whose dynamics evolve in continuous time.

Various factors may underlie a tendency toward unstable behavior in a controlled, coupled system such as the virtual wall. These include non-collocated sensor and actuator [Eppinger and Seering 86], system dynamics which are unmodeled or otherwise omitted from the controller design, and signal quantization. In the field of robotics, numerous controllers and design directives have been developed for achieving robust transition from unimpeded motion to contact with a workpiece [Hyde and Cutkosky 93]. In the field of haptic display, we have the luxury of being able to make use of what has been learned in robotics and thus avoid many of the destabilizing mechanisms. Two destabilizing mechanisms, however, have not been treated extensively in robotics and are not yet easily side-stepped by informed design. These are the zero-order-hold operator and the possible asynchrony of wall threshold crossings with sampling times. Both of these processes are inevitable consequences of the sampled data implementation of the virtual wall. Note that interest in the treatment of sampling delays and intersample effects in robotics has been small since fast sample rates are generally available. Haptic display algorithms, on the other hand, usually compete with graphical updates for computational resources. Furthermore, a very perceptive and critical human user is involved. The desire to use stiff walls in graphically interesting virtual environments creates the need for computationally efficient yet stable wall algorithms. We surmise that the existing handicap on virtual walls must be removed since competing demand for computational resources will certainly continue to grow faster than available supply. This paper addresses chatter associated with stiff virtual walls by developing improved virtual wall controllers. These virtual wall algorithms, when used in the standard sampled data implementation for haptic display, will render walls which do not suffer chatter even

when the wall stiffness is high and the sampling period long.

Energy Leak: The Zero Order Hold

From digital control theory, a controller designed in the continuous domain will yield poor results if implemented as a sampled-data controller when the sampling period is long, especially if the plant possesses fast dynamics. The nominal half-sample delay associated with the zero order hold will act to destabilize the system. In the case of a stiff virtual wall, fast dynamics can indeed be expected. An intuitive explanation is also available for the energy instilling effects of the zero-order hold. (See discussions in [Colgate et al.]) Suppose the standard wall implementation is used, where a reaction force is computed as the product of stiffness and sampled wall penetration. While moving into a wall, the sampled manipulandum position will be closer to the wall surface than the manipulandum itself and thus the force of the wall will be too low (compared to a real wall). By contrast, while moving out of the wall, the sampled position will lie deeper inside the wall than the actual manipulandum position and as a result the force will be too high. Thus as one presses on the virtual wall, one needs to perform less work than one would on the real wall to produce the same deformation. As one lets go, one has more work returned by the virtual wall than would have been returned by its real-world counterpart. Thus to simply push on the wall and let go (a common exploratory procedure) is an effective method for extracting energy from the virtual wall.

Several papers have undertaken analyses of sampled-data delays in the virtual wall in order to delineate regions in parameter space that lead to suitable wall implementations. The destabilizing effects of delay were addressed in [Minsky et al. 90], although the delay was attributed to computation rather than the zero order hold operator and human limb dynamics were neglected. Kazerooni analyzed the effects of time delay in [Kazerooni 93] and suggested lower bounds on the sampling rate to avoid contact instability. Love and Book used the Jury stability criterion to analyze contact instability in virtual walls while assuming a constant force applied by the human user [Love and Book 95]. Most notably, in order to circumvent the need to model and analyze the human, Colgate and Brown have delineated the regions in parameter space which lead to *passive* [1] virtual walls [Colgate et al.] and [Colgate and Brown 94]. So long as the user maintains a strictly passive impedance, the coupled system is guaranteed to remain stable by the passivity theorem [Colgate and Schenkel 94] and [Colgate 94]. Colgate's passivity analysis yields a valuable set of design guidelines as well. An implementation of a virtual wall must contain some inherent physical damping if it is to behave passively. Negative virtual damping (derived from a low pass filtered force signal) may be used in the controller to compensate for the added physical damping. Colgate's passivity analysis and resulting design guideline constitute an elegant solution to the destabilizing effects of the zero order hold since human limb dynamics need not be considered explicitly. Insofar that the human limb may be considered dissipative, however (see below), the passivity approach is conservative.

In this paper, we adopt a stability rather than a passivity analysis and model the human user dynamics explicity. This approach allows us to go beyond delineating regions in parameter space and actually eliminate energy leaks for all parameter values. Rather than avoiding the problem, we develop virtual wall design methodologies which *solve* the problem at its root —for all regions in parameter space.

Energy Leak: Asynchronous Switching Times

A more subtle energy-instilling aspect of the sampled data implementation of the virtual wall has to do with the fact that the wall uses a switching controller. Briefly, this effect is due to asynchrony of traversals of the wall threshold with the sampling times. Due to discrete sampling, control law changes are not enacted until the next sampling time after the crossing of a threshold. Crossings of the threshold which occur between sample times can effectively introduce energy as follows.

Figure 1 shows the trace of the zero-order held output force f_k resulting from a single strike of a unit-stiffness wall overlaid on top of a time trace of the manipulandum position y. The wall is positioned at $y = 0$. The sampling period is 0.1 seconds. The on-average small valued force on the way into the wall and large valued force on the way out of the wall can be seen in Figure 1 (recall the zero-order hold discussion above). But two time periods in particular are highlighted. The first, labeled Δt_a, is the delay in turning on the wall controller and Δt_b is the delay in turning off the wall. Observe that the wall will first be turned on with the wall's spring in a slightly compressed state because the first sampled position to trip the conditional will be located inside the boundary. Such an error can produce energy since the spring now stores energy without the requisite work having been done. Upon leaving the wall, the first sampled manipulandum position will, in all likelihood, lie outside of the wall rather than just on the threshold. In this case, since the wall will immediately be turned off (assuming no computational delays), the force for the last wall-on

[1] We adopt the standard definition of passivity, that energy cannot be extracted from a passive system. Technically, for all possible force/motion trajectories, the running integral of force times velocity is always less than the initial stored energy.

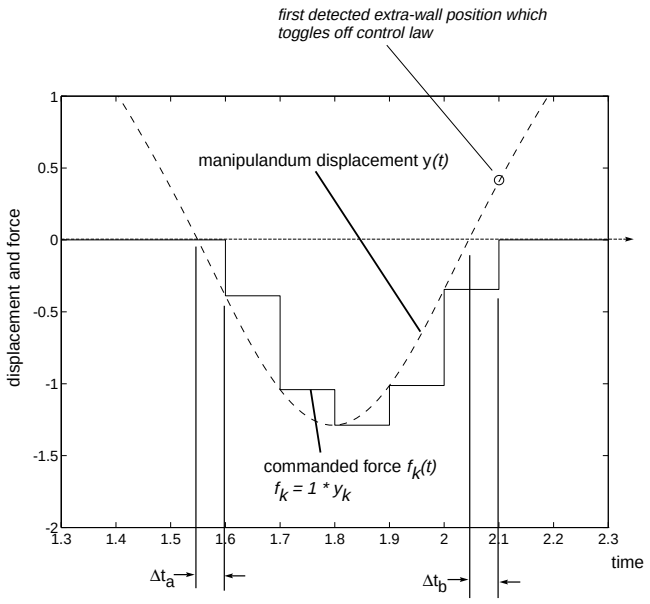

Figure 1: *Time-chart of modeled manipulandum position and control effort*

sample period will still push away from the wall. Thus we see that the asynchrony of the wall on/off switching times with the sampling times has a net energy- instilling effect.

Role of the Human in the Interaction Dynamics

The tendency toward chatter depends to a large extent on the physical properties of the human user—specifically, the human's driving point mechanical impedance at the interface. (Throughout our treatment, we assume that the human's finger maintains contact with the manipulandum.) Under consideration is the interaction behavior of two dynamical systems, the manipulandum and the human limb. But even further, under consideration is the interaction behavior of two *controlled* dynamical systems. Behavioral predictions cannot be made until both systems, each with their controller are brought into the analysis. Note that the driving point impedance of the human hand or finger can be modulated (within bounds) by the human user by changing muscle activation levels or hand/finger postures. Thus, by pressing in certain ways, chatter against a virtual wall can be selectively induced and sustained, and sometimes even amplitude-modulated. Another interesting empirical observation to be made regarding walls is that the same wall may be destabilizable (prone to chatter) under the fingers of one person while always remaining stable under the fingers of another.

The foregoing examples highlight the way in which chatter is usually encountered and points to a modeling assumption which can be used to greatly simplify the analysis (and design) of the virtual wall—that is to assume constant command on the part of the human. Chatter frequencies are typically on the order of 10-50 Hz. An effective strategy for the human to induce chatter is usually not to move back and forth at high frequency but rather to adopt and maintain a certain impedance while simply hitting the wall a single time (or even gently coming up against the wall). Although not always beyond the command capabilities of the human, typical chatter frequencies are certainly high compared to the frequencies which characterize the wall-strike *intentions* of the human. Hannaford and Anderson [Hannaford and Anderson 88] demonstrate the existence of contact instability in simulation of hard contact through a bilateral teleoperator when a time-invariant sixth order model is used for the human and sampling delays are included. This chattery behavior closely followed experiment and also showed a dependence on user grasp.

Armed primarily with the observation that chatter remains, both empirically and in simulated settings, when a constant impedance is adopted by the human, we assert that the human is not responsible for introducing energy into the system. Further, the literature indicates that second order linear models may be used to model a human finger to a very good degree of fit so long as the time durations are short enough to preclude volitional control and reflexes (about 30 ms) [Hajian and Howe 94]. We will therefore assume that the wall-exploring human may be fit with a second order linear time-invariant model. We propose that models of the human finger may be constructed using force and motion data collected with the same haptic interface hardware which is used for display, just prior to rendering of the wall.

Outline

Having identified two mechanisms underlying the non-passive behavior of virtual walls, and having in hand a model of the full coupled system, we embark on the design of improved (compensated) virtual wall controllers.

We shall draw from the work of Howe [Howe 84], who presents a technique involving model-based prediction to account for sampling delays in real-time flight simulation with motion display. We extend Howe's ideas to include the human user, which amounts to an inclusion of feed-through dynamics in flight simulation. We shall develop an intersample threshold crossing effect watchdog and recover from devia-

tions which that watchdog informs us about through an application of deadbeat control. A related problem arising in simulation through discontinuities with a constant step-size algorithm was treated by Lin and Howe [Lin and Howe 91]. The inclusion of a possibly varying human impedance in our system, however, precludes our use of an off-line simulation and look-up table approach as was proposed by Lin and Howe. We rather adopt a watchdog-report-correct approach.

In the following section, the model of a bouncing ball which will serve as a useful allegory for the development of the controller designs is presented. Thereafter, the design of two improved virtual wall algorithms will be carefully developed. The first design uses model-based prediction, the second design makes use of standard state-space digital control design techniques. The asynchrony watchdog and deadbeat control methods will also be introduced. The final section will discuss preliminary experiments, summarize, and discuss extensions.

MODELING THE SAMPLED DATA SYSTEM

To expound the controller designs, a representative discontinuous system will be discussed: a suspended ball bouncing on an elastic floor. The bouncing ball is used as a model of human exploration of a virtual wall through a haptic interface and as a kind of work-bench for the design of virtual wall controllers. Figure 2 shows a bouncing ball

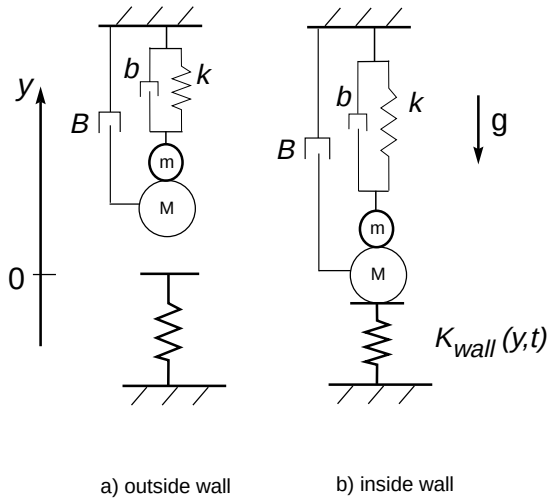

Figure 2: *Modeling the Bouncing Ball*

in two configurations. The ball has mass $m+M$ and is suspended from the ceiling by a spring of stiffness k and two dampers of damping coefficient B and b. Parameters k, b and m pertain to the human finger impedance. Parameters M and B pertain to a damped mass model of the manipulandum. [2] The configuration in Figure 2 a) corresponds to the *outside wall* condition. Here, the ball is being acted on solely by the force of gravity and the spring- damper forces. In Figure 2 b), corresponding to the *inside wall* condition, the ball is being acted on by the force of gravity, the spring-damper forces, and the force of a special spring $K_{wall}(y,t)$ depicted as an elastic floor in Figure 2. The rest position of the floor is taken to be zero ($y = 0$) which is consistent with the models fit by Hajian and Howe [Hajian and Howe 94]. The gravity constant can be varied to model various bias forces applied by the user.

In order to allow for the subsequent reinterpretation of the 'floor' as a virtual wall controller in a sampled-data system, the floor simulator is specially structured as follows. Simulations of the floor are allowed to communicate with simulations of the ball only at certain time points (the sampling times). The switching times are also constrained in this simulation to lie on the sampling times. Furthermore, the force response of the floor shall be held constant between sampling times to depict the zero-order-hold. The floor is thus simulated as a discrete system and the ball as a continuous system. While the ball depicts the manipulandum and human (plant), the floor depicts the discrete controller, which we shall henceforce refer to as the "wall". Figure 3

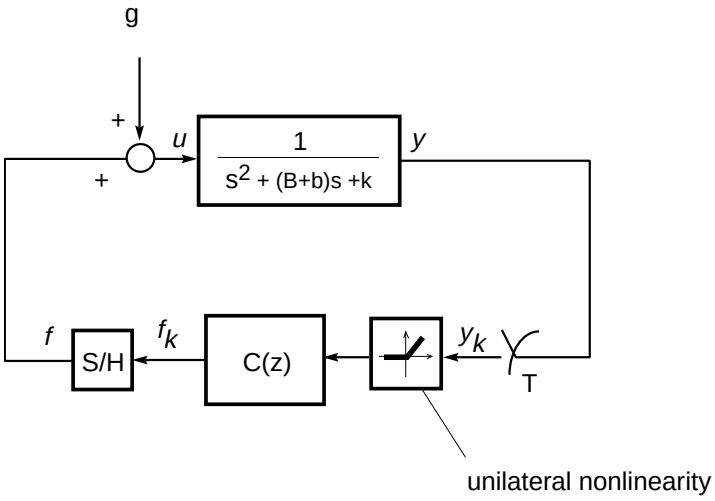

Figure 3: *Sampled Data System-inspired Block Diagram for the Bouncing Ball Simulator*

shows this discrete wall as the feedback controller in a block diagram with the ball as plant. The discrete wall controller $C(z)$ is shunted into the loop only during the *inside wall* periods of simulation by the unilateral nonlinearity.

[2] We assume that the manipulandum and finger are rigidly attached. Such an assumption is valid so long as the finger pulp is sufficiently compressed, which occurs for forces greater than about 2N [Hajian and Howe 94].

Sampled-data Bouncing Ball Simulation

We begin with a wall control law for simulation like that most commonly used in virtual walls (the control law inspired by its continuous time counterpart), namely $f_{wall} = K_{wall} * y_k$. The differential equation (model) for the suspended mass is simply:

$$(M+m)\ddot{y} + (B+b)\dot{y} + ky = -g - f_{wall} \qquad (1)$$

Note that the reaction force of the wall is a forcing term (on the right-hand side) in this model. The motion of the ball is simulated in intervals, each the length of one sampling period T, with an ODE solver. At each sampling point (between intervals) an indicator function is checked (whether the ball is inside the domain of the wall). If the indicator function evaluates true, the reaction force of the spring, f_{wall}, is computed according to the control law and held constant for the duration of the next sampling period. If the indicator function evaluates false, f_{wall} is set to zero.

Continuous Bouncing Ball Simulation

In order to produce the motion of a continuous bouncing ball for comparison, two models are sequenced together in simulation: one of the suspended mass (Eq. 1) with $f_{wall} = 0$, the other of the suspended mass with an additional spring of stiffness K_{wall} incorporated in place of the forcing term f_{wall}:

$$(M+m)\ddot{y} + (B+b)y + (k+K_{wall})y = -g \qquad (2)$$

Sequencing between these two models in the case of the continous switching system takes place strictly as a function of threshold crossing times, without sampling effects. Appropriately, no zero order hold is modeled with the inclusion of the wall spring forces on the left hand side of Eq. 2. Accurate sequence timing can easily be determined without simulation backstepping if each model can be set up to return the time remaining to a threshold crossing from a given state. Unfortunately, the time remaining to a threshold crossing from a state either in the *outside wall* (Eq. 1) or *inside wall* (Eq. 2) models cannot be computed analytically so long as both the bias force (gravity) and damping are included ([Gillespie 96], p. 170). To resolve this issue and also to simplify the exposition here, we shall assume negligible damping in both the manipulandum and human limb.

Without damping, the solution to the *outside wall* model (Eq. 1) reads:

$$y(t) = C_1 cos(\omega_1 t) + C_2 sin(\omega_1 t) - g/w_1^2 \qquad (3)$$

where

$$C_1 = y_0 + g/w_1^2$$
$$C_2 = v_0/w_1$$
$$w_1 = \sqrt{k/(m+M)}$$

The solution to the *inside wall* model (Eq. 2) reads similarly, except that $w_2 = \sqrt{(k+K_{wall})/(m+M)}$ is used in place of w_1. The first root of Eq. 3, or time Δt_1 which yields $y = 0$ is given by:

$$\Delta t_1 = \frac{1}{w_1}\left[\pi + atan2(C_1, -C_2) + sin^{-1}\left(\frac{g/w_1^2}{\sqrt{C_1^2 + C_2^2}}\right)\right] \qquad (4)$$

where $atan2$ is the four-quadrant arc tangent function. The root of the *inside wall* model solution Δt_2 reads similarly, with ω_2 substituted for ω_1.

An ODE solver can be set up to sequence between simulation of the *outside wall* (Eq. 1) and *inside wall* (Eq. 2) models using time periods given by Eq. 4. Alternatively, the solutions themselves (Eq. 3 with ω_1 and ω_2) may be used with an indicator function and backstepping.

Simulation Results

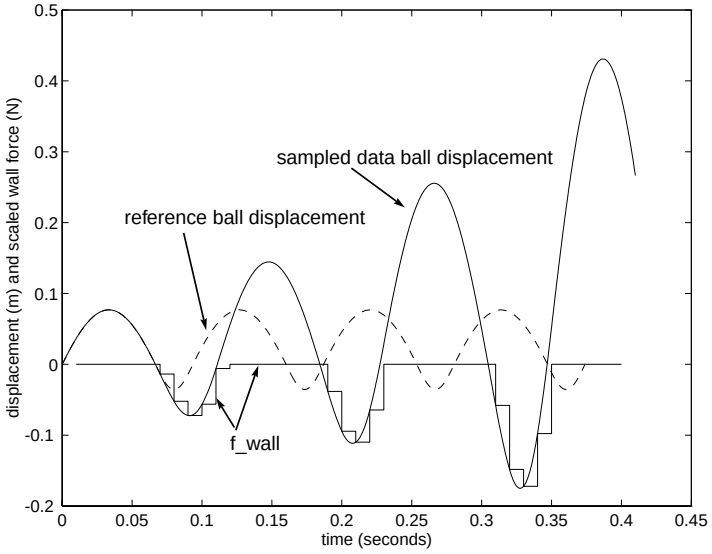

Figure 4: *Sampled Data Algorithm Simulation Results*

Figure 4 shows the simulation results using the "Sampled Data Bouncing Ball" simulator along with the results of the "Continuous Bouncing Ball" simulator for reference, drawn with a dashed line. The staircase-shaped trace of the reaction force, f_{wall} is also plotted, scaled by the in-

verse stiffness $1/K_{wall}$. Parameter values used in these and subsequent simulations are given in Table 1.

Table 1.

Parameter	Symbol	Value	Units
Time Step	T	.01	sec
Bias Force	g	50	N/s^2
Wall Stiffness	K_{wall}	5000	N/m
finger stiffness	k	500	N/m
finger damping	b	0	N-s/m
finger mass	m	0.006	kg
Manip. damping	B	0	N-s/m
Manip. mass	M	0.35	kg

We quickly note that the sampled data wall produces non-physical behavior; the ball bounces higher and higher. Given that we have assumed no damping, yet the ball bounces higher, we have found evidence of an energy leak —the destabilizing effects of the zero order hold and of intersample threshold crossing. Inclusion of damping would produce a limit cycle behavior, where energy introduced is balanced by that dissipated. So the immediate goal which directs the controller design is simply to eradicate this 'non-physical' simulated behavior. Upon coming up with a suitable wall simulation (one in which the lossless ball attains the same height each bounce), that simulation algorithm may be reinterpreted as a wall controller and implemented directly in the actual physical hardware. At this point, the design of improved controllers is underway.

CONTROLLER DESIGN

Two controller designs will be presented, the first based on half-sample prediction and the second on design in the digital domain. Both of these controllers are intended to compensate for the destabilizing effects of the zero order hold. An enhancement to both of these controllers, which compensates for the effects of intersample threshold crossing, will be presented in the third part of this section.

Half Sample Prediction Controller

Our first improved controller is inspired by noting that the effect of the zero order hold can be approximated by a half-sample time delay. An improved controller will be constructed by adding half-sample prediction to the algorithm with the aim of cancelling the effect of the zero order hold.

We already have a model of the target system (a suspended mass) in hand in the form of Eq 2. At each sample time $t = kT$, we can simulate ahead using this model or, even easier, evaluate its solution (Eq. 3 with ω_2) at $t = kT + T/2$ and starting from the present state $[y_0 v_0] = x(kT)$ to predict the ball's position a half sample ahead. This predicted position is then used in the standard control law $f = K_{wall} * y$.

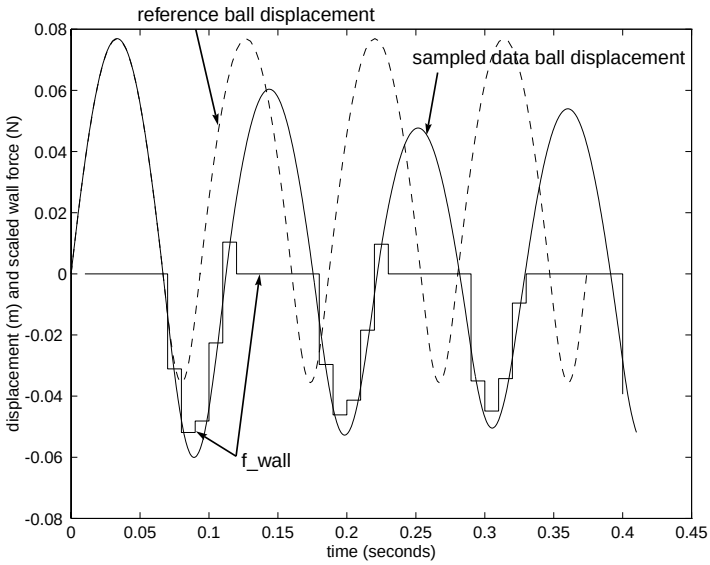

Figure 5: *Half sample prediction simulation results*

Figure 5 shows the simulation results of the above half-sample prediction algorithm. Again, the wall force has been scaled by the inverse stiffness $1/K_{wall}$. Here one sees how the trace of the zero-order held spring force intersects the continuous wall position trace approximately midway between sample times. This propitious intersection leaves half of the inscribed area above and half below in contrast to the staircase plot which the reader may recall from Figures 1 and 4. Our new algorithm does not climb "uphill both ways".

The other interesting thing to note in Figure 5 is that occasionally, for the last *inside wall* sample period, the wall is actually exerting a tensile force, pulling on the user. During this period, since the manipulandum is moving in the direction away from the wall, the user is doing work on the wall. During motion away from maximum wall penetration, the wall is for the most part returning work to the user, except for this last sample period.

Though the half-sample prediction method outlined above yields vastly improved results over those shown in Figure 4, the results are still not perfect. Deviations from the desired bouncing path can already be seen in Figure 5, but excursions become especially apparent if the algorithm is allowed to continue for some time. The bouncing height is irregular, sometimes higher, sometimes lower than the target height.

Reasons for this erratic behavior are twofold. Firstly, the ZOH is only approximated by a half sample de-

lay; its full effect is more complex. See, for example [Franklin and Powell 90]. The next section will present a design using digital controller design tools which fully accounts for the effects of the ZOH. Secondly, the wrong control law will be used to compute the reaction force for certain portions of those sample periods which contain the threshold crossings, thus exerting a force inappropriate to that portion of the time period. Stated another way, turning on and turning off of the wall control law will not necessarily occur when the ball height is $y = 0$. A fix for this second phenomenon which we call intersample threshold crossing will be presented in the section following the next.

Design in the Digital Domain

Another approach to the design of a controller for our bouncing ball simulator (and in turn for the haptic display of a virtual wall) is controller design in the digital domain. Our interim goal is to design a controller for a discretized plant, such that the response of the closed loop discrete system is the same as the desired (continuous bouncing ball) response on the sampling times.

First the ZOH discrete equivalent of the desired dynamics (a mass without damping sprung by both k and K_{wall}) are found using a table of $\mathcal{Z}$ transforms.

$$\mathcal{ZOH}\left\{\frac{1}{s^2+\omega_2^2}\right\} = \frac{(1/\omega_2)(1-cos\omega_2 T)(1+z)}{z^2-(2cos\omega_2 T)z+1} \qquad (5)$$

Where $\mathcal{ZOH}\{\cdot\}$ denotes the zero-order-hold discrete equivalent of the bracketed expression. Note that $\mathcal{ZOH}\{\cdot\} = (1-z^{-1})\mathcal{Z}\{\cdot\}$.

This discrete equivalent has two complex poles and one zero. The pole locations (roots of the characteristic equation), λ_1 and λ_2, are identified as the desired root locations for the closed loop system. These root locations correspond to the response of the referent system, a sprung mass, expressed in the digital domain.

We will perform the design of our controller in the digital domain. For this purpose, the ZOH discrete equivalent of the plant $\frac{1}{s^2+\omega_1^2}$ is found.

$$\mathcal{ZOH}\left\{\frac{1}{s^2+\omega_1^2}\right\} = \frac{(1/\omega_1)(1-cos\omega_1 T)(1+z)}{z^2-(2cos\omega_1 T)z+1} \qquad (6)$$

Using full state feedback, the poles of this system may be placed arbitrarily. We simply choose to place the roots of this controlled system at the locations λ_1 and λ_2 using pole placement. The full state feedback gain $K = [k_1 k_2]'$ which places the closed loop dynamics of the sytem at these target locations is available from a pole placement algorithm such as Aizerman's method.

The response using this controller is nearly identical to that of the previous section using the half-sample prediction controller. Once again, the system is still not perfect, due to errors in switching the controller at the wall threshold (intersample threshold crossing). The next section will address this very problem, producing results from the sampled data controller which are indistinguishable from the continous target system.

Compensation for Asynchrony

The effects of inter-sample threshold crossing can be fully accounted for by using solutions (or simulations) of the *outside wall* and *inside wall* models. For clarity, the approach is first outlined: First, we deduce the inter-sample entry and exit times from information available at the sampling times. We then predict, again using the model solutions, what state the ball would be expected to attain, if the switching were on-threshold, at the first sampling time outside of the wall. Finally, we drive it to that desired state (as shown below) with the last two zero-order-held force values inside the wall.

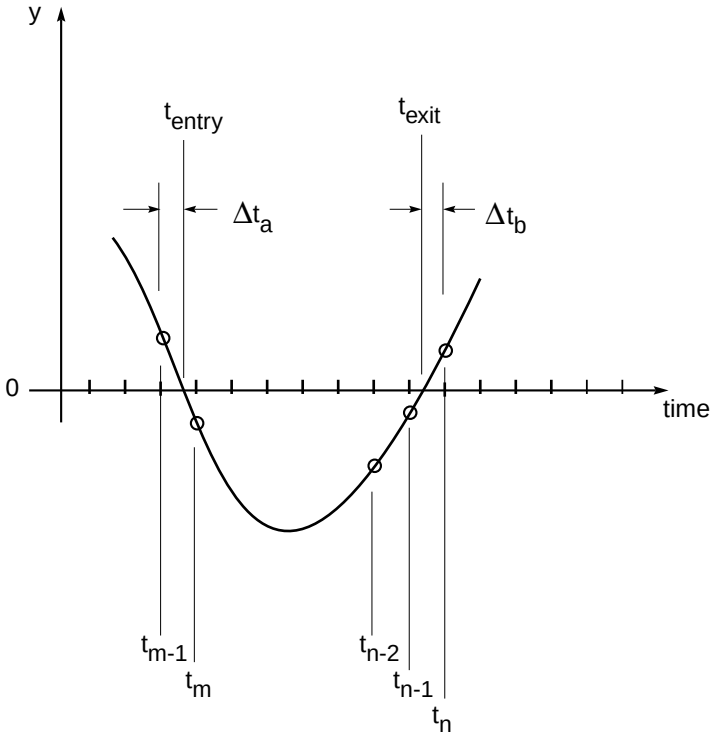

Figure 6: *Sampling Points in a Typical Floor Strike*

Figure 6 shows an arbitrary placement of sampling times

on the motion path of a simulated strike of the wall. The first sampling time for which the ball is inside the wall is designated t_m and the first outside of the wall is designated t_n. Reference to these time points will be made in the following discussion.

The state of the ball at the first sampling time outside of the wall, $x(t_n)$ encapsulates the action of the wall simulator. The state of the ball at t_n resulting from an on-threshold switching wall is obviously not the same as the state resulting from either the standard algorithm wall (see Figure 4) or the improved half-sample prediction wall algorithm (see Figure 5). But, having assumed models for both the *inside wall* and *outside wall* conditions, the state of the ball resulting from an on-threshold switching wall which we shall call x_d, (d for *desired*) can be extracted from the state information available on the sampling times. To find this state is a multi-step process as follows:

First, the root of the *outside wall* model is found, using the known state at the last sampling time before entry $x(t_{m-1})$ in Eq. 4 (with ω_1). This time interval is designated Δt_a (see Figure 6).

The full state at wall entry $x(t_{entry})$ is found using the solution to the *outside wall* model (Eq. 3) and its derivative by evaluating these at $t = \Delta t_a$ and $[\ y_0 \quad v_0\ ]' = x(t_{m-1})$.

Now the time at which the exit from the wall is made, t_{exit}, is found using the initial condition $[\ y_0 \quad v_0\ ]' = x(t_{entry})$ in the root of the *inside wall* model (Eq. 4, with ω_2).

Note that the state at t_{exit} is already known, quite simply, because it is an undamped wall:

$$x_{exit} = \begin{bmatrix} y_{entry} \\ -\dot{y}_{entry} \end{bmatrix} \tag{7}$$

where the shorthand x_{exit} stands for $x(t_{exit})$. For models which include damping, the exit state could be obtained using Δt_1 and the solution to Eq. 2.

Knowing the time of exit t_{exit}, the time remaining to the first sampling period outside of the wall Δt_b (see Figure 6) is available:

$$\Delta t_b = t_n - t_{exit} \tag{8}$$

where $t_n = kT$ and k is the smallest integer such that $y(kT) > 0$ since the last wall encounter.

Finally the desired state at time t_n can be computed by evaluating the solution to the *outside wall* model (Eq. 3 with ω_2) starting at the exit state $[y_0 v_0]' = x_{exit}$ and $t = \Delta t_b$. This state $x(t_n)$ we shall call the desired state x_d.

We now know where this system is to be driven so that it will in the end behave as the continuous bouncing ball. The problem has become how to shape the control force f_{wall} so that, at t_n, the state will arrive at x_d. This is once again a problem of controller design in the digital domain. We start by discretizing the continuous part of the system, which is of course only seen by the discrete controller at the sampling times kT. We find the ball's zero-order hold equivalent $\{\ \Phi \quad , \quad c\ \}$ so that the effects of the zero-order hold are included in its discrete representation. Given that the discretized system is only second order and that it is fully controllable, it will only take two steps to drive it to any desired state.

The response of a discrete system $\{\ \Phi \quad , \quad c\ \}$ to the input sequence $u(kT)$, $k = 0, ...n$ can be expressed:

$$x(n) = \Phi^n x(0) + \mathcal{C} \begin{bmatrix} u_{n-1} \\ u_{n-2} \\ \vdots \\ u(0) \end{bmatrix} \tag{9}$$

Where the controllability matrix $\mathcal{C}$ is given by:

$$\mathcal{C} = [\ , \quad \Phi, \quad \ldots \quad \Phi^{n-1}, \ ] \tag{10}$$

Given controllability ($det(\mathcal{C}) \neq 0$), this equation can be inverted for the control sequence

$$\begin{bmatrix} u_{n-1} \\ u_{n-2} \\ \ldots \\ u(0) \end{bmatrix} = \mathcal{C}^{-1}(x_d - \Phi^n x(0)) \tag{11}$$

Since our system is only second order, it will only take two inputs to drive the system to state x_d:

$$\begin{bmatrix} u_{(n-1)} \\ u_{(n-2)} \end{bmatrix} = \mathcal{C}^{-1}(x_d - \Phi^2 x0) \tag{12}$$

where $x0$ is the state $x(t_{n-2})$, two samples before the first sample outside of wall, with n pertaining to Figure 6. The control value u_1 is used at t_{n-2} and u_2 is used at t_{n-1}. [3]

The simulation results using this watchdog and deadbeat control algorithm are shown in Figure 7. After each application of deadbeat control, the displacement of the compensated sampled data system is indistinguishable from the

[3] Deadbeat control typically refers to a design in which full state feedback is used to place all eigenvalues at the origin so that with no input, the state reaches zero in m steps, where m is the system order. We apply the term 'deadbeat' to our design solution since only 2 input steps are used to drive this second order system to state x_d.

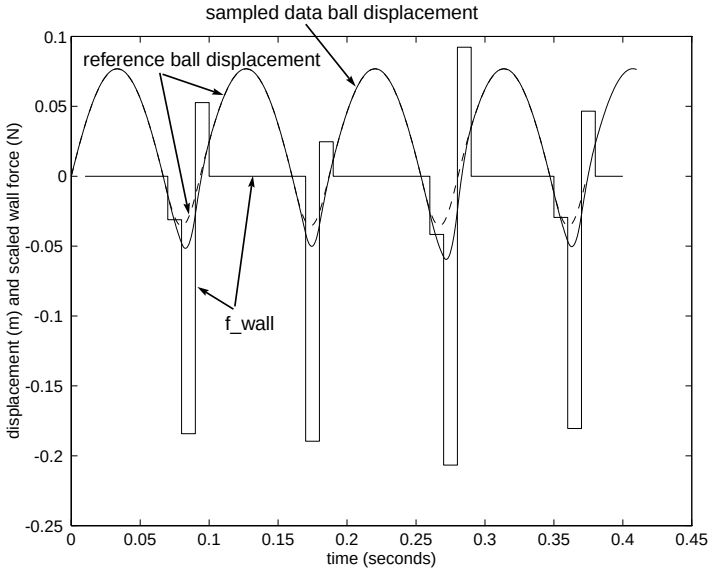

Figure 7: *Full Control Algorithm Simulation Results*

displacement of the on-threshold switching system. The large excursions in the new virtual wall force f_{wall} are due to the significant deviation from the refernt path, which in turn has to do with the fact that wall penetration lasts for only 3 samples for the particular wall parameters used here. The parameter values from Table 1 represent a somewhat extreme case, chosen to produce readable plots.

EXPERIMENT AND DISCUSSION

The half-sample prediction controller described in the forgoing section was coded in C++ and tested experimentally. The new virtual wall controllers performed significantly better than the old in side-by-side tests. Walls rendered with the new controllers did not support sustained oscillations for those parameter values under which the old virtual walls in fact did support sustained oscillations. Experimental results, however are sill forthcoming and will be reported in future publications.

The forgoing controller exposition has concentrated solely on the rendering of *undamped* virtual walls. Yet the new controllers use full state feedback, with gains on both sensed position and sensed velocity. It is not really fair to compare a wall control law which uses both position and velocity feedback gains with wall controllers which only use position feedback gains. Comparisons between the new controllers and *damped* virtual walls would be more appropriate —especially since damping is often added to virtual walls to enhance their stability, but usually by trial and error. Indeed, this trial and error process in the design of damped walls makes their direct comparison difficult. The significant improvements exhibited by the new controllers can be attributed to the addition of positive velocity feedback.

The point of the new controllers, then, is not so much their highly improved performance as promulgated above, but rather that a method for designing feedback gains has been presented which, when implemented in the sampled data setting, will exhibit the desired stiffness. Damped walls with desired stiffness *and damping* can also be designed with the same method. The model to be used in the half-sample prediction controller would then be a damped second order oscillator rather than the undamped oscillator used in the exposition of this paper. The desired pole locations in the design-in-the-digital-domain procedure would turn up inside the unit circle. The same Aizerman pole placement algorithm would produce appropriate feedback gains on position and velocity just as in the undamped case shown. The risk of coming up with a wall which feels damped because excessive damping was chosen (providing more stabilizing influence than necessary) is not a problem with these new design techniques.

These comments pertain to the compensation for the zero order hold. The dead-beat control techniques may of course also be used in the design of damped walls. This technique for the quelling of effects from intersample threshold crossing does not have precedent in the literature.

SUMMARY AND EXTENSIONS

This paper has addressed the formulation of controllers designed to create the illusion, for a human user, of a passive wall which opposes motion with spring forces when the user drives a manipulandum past a threshold. A set of virtual wall controllers has been presented which are immune to two certain destabilizing factors which otherwise play a significant role in the implementation of virtual walls, causing chattery behavior. Both of these factors are a consequence of the sampled data setting within which virtual wall controllers must operate. First, the zero order hold effectively introduces loop delays which, under position feedback (as called for by a stiff wall) introduce energy into the closed loop system. The effects of the zero order hold are quelled in the new virtual wall controllers using two alternate design techniques: half-sample prediction and design in the digital domain. Both of these controller enhancements were shown to substantially improve virtual wall performance (decrease chatter) in simulation. Close scrutiny of simulation, however, shows that something is still amiss in these sampled-data virtual walls: the issue of intersample threshold crossing, the second factor which we address.

The controller for the virtual wall is a *switching* con-

troller. Because the on/off switching is based on a signal which is only discretely sampled, timing errors are introduced into the switching behavior of this controller. The effects of intersample threshold crossing may be fully counteracted, however, by model-based deduction of the timing errors from state information collected on the sample times and with an application of dead-beat control. The fact that the controller is discrete works to our advantage this time, since deadbeat control is able to perfectly compensate for the errors of intersample threshold crossing and deadbeat control is *only* available in discrete controller implementations.

Perhaps the most noteworthy aspect of these new controllers is that, in their design and operation, a simple time-invariant model of the human user has been assumed and utilized. Justification for this somewhat bold move was drawn from fact that the dynamics in question are outside the range of voluntary movement for humans. Further backup is provided by system simulations which model the human input as a constant bias force that exhibit the same chattery behavior which is empirically observed in the old (un-improved) virtual wall controllers.

We have introduced the incorporation of mechanical impedance properties of a person's finger or limb in controller designs for the particularization of the rendering of a virtual haptic environment. Guaranteed performance and optimum perceptual fidelity of synthesized mechanical properties are now possible. We look forward to on-line identification of the human mechanical impedance so that, when a user changes limb posture, properties of the haptic interface would continue to provide maximum impedance- range yet guaranteed passive behavior. On-line system identification of the user would accomodate changes in posture or muscle-activation chosen by the user. Auspiciously, these time-scales are much slower than the time-scales of the energy leak mechanisms and the compensation which depends on those user models.

Other natural extensions to the method include the following: Computational delays can also be compensated out using model-based prediction or design in the digital domain. Lookup tables can be substituted for the functions mentioned above to facilitate speed (ease computational overhead). More complex kinematic constraint changes such as those arising from collisions in dynamical system simulation could be handeled by a fixed step size integrator. The deadbeat control techniques could be used to drive a system to a known appropriate state when that known state is derived by methods such as examnimation of an integral of the motion, energy conservation, or sensed power exchanges with the user. The correcting controls could be spread over more than two steps or placed more often in the algorithm to minimize control excursions. Finally, an analysis of the effect of using a first-difference estimate for velocity within these algorithms would be very valuable.

[Colgate 94] J. E. Colgate. "Coordinate transformations and logical operations for minimizing conservativeness in coupled stability criteria," *Journal of Dynamic Systems, Measurement and Control*, vol. 116, no. 4. pp. 643-649, December 1994.

[Colgate and Brown 94] J. E. Colgate and J. M. Brown. "Factors affecting the Z-Width of a haptic display," In: *International Confrerence on Robotics and Automation*, December 1994.

[Colgate et al.] J. E. Colgate, P. E. Grafing, *et al.*. "Implementation of stiff virtual walls in force-reflecting interfaces," In: *Virtual Reality Annual International Symposium*, Seattle, WA, pp. 202-208. September 1993.

[Colgate and Schenkel 94] J. E. Colgate and G. Schenkel. "Passivity of a class of sampled-data systems: application to haptic interfaces," In: *American Control Conference*, pp. 3236-3240. June-July 1994.

[Colgate and Stanley 95] J. E. Colgate and M. C. Stanley, *et al.* "Issues in the haptic display of tool use," In: *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, pp. 140-145. 1995.

[Eppinger and Seering 86] S. D. Eppinger and W. P. Seering. "On dynamic models of robot force control," pp. 29- 34. 1986.

[Franklin and Powell 90] G. F. Franklin, J. D. Powell, *et al. Digital Control of Dynamic Systems*. Addison Wesley, 2nd edition, 1990.

[Gillespie 96] R. B. Gillespie. *Haptic Display of Systems with Changing Kinematic Constraints: The Virtual Piano Action*. Ph.D. thesis, Stanford University, 1996.

[Hajian and Howe 94] A. Z. Hajian and R. D. Howe. "Identification of the mechanical impedance of human fingers," submitted to: *The ASME Journal of Biomechanical Engineering*. See also: *ASME Conference and Exposition, Dynamic Systems and Controls*, vol. 1 no. 55, 1994.

[Hannaford and Anderson 88] B. Hannaford and R. Anderson. "Experimental and simulation studies of hard contact in force reflecting teloperation," In: *International Conference on Robotics and Automation*, vol. 1, pp. 584-589. 1988.

[Howe 84] R. M. Howe. "Techniques for optimizing computer performance in real-time flight simulation."

[Hyde and Cutkosky 93] J. M. Hyde and M. R. Cutkosky. "Contact transition control: an experimental study," In: *International Conference on robotics and Automation*, vol. 1, pp. 363-368, May, 1993.

[Kazerooni 93] H. Kazerooni. "Human induced instability in haptic interfaces," In: *ASME Winter Annual Meeting, Dynamic Systems and Controls* Vol. 55-2, pp. 851-856. 1993.

[Lin and Howe 91] K. C. Lin and R. M. Howe. "A new approach to the real-time simulation of control systems with discontinuities," In: *American Control Conference*, vol. 2, pp. 2122- 2127. June 1991.

[Love and Book 95] L. Love and W. Book. "Contact stability analysis of virtual walls," In: *ASME Conference and Exposition, Dynamic Systems and Controls*, vol. 57-2. pp. 689-694. 1995.

[Minsky et al. 90] M. Minsky *et al.* "Feeling and seeing: issues in force display," In: *Computer Graphics: ACM Transactions on Computer-Human Interactions*, vol 24-2, pp. 235-243. March 1990.